REMARKS

Claims 1-24 are pending in the present application. Claims 1-3, 9-11, and 17-19 were amended. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 101

The examiner has rejected claims 1, 9, and 17 (and, necessarily, their dependent claims) under 35 U.S.C. §101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed. The independent claims have now been amended to recite that a possible hard-coded string is "flagged" after it is identified. One of ordinary skill in the art would understand that to flag an item is to make some type of mark or indication that sets this item apart from similar items. In computer arts, it would be understood that flagging a string would require adding an indicator to a stored or printed version of the string so that the item is easy to pick out again. Thus, one of ordinary skill would recognize that flagging a string is a tangible change that reflects the intangible recognition of the contents of the string.

The rejection further stated that no practical application had been given for the disclosed invention. However, it is submitted that one of ordinary skill in the art would recognize, from the background information, the practical application of the described method. The background section of the application give the problem that is being solved, from which one of ordinary skill would know the practical application for the inventive process and system. That is, the application notes that,

> "Internationalization is a process of enabling a program … to run correctly in any country. … to read, write and manipulate localized text. … conform to local customs when displaying dates and times …Various scanning programs have been developed which attempt to detect hard-coded strings [so that they can be corrected]. Unfortunately, these scanning programs simply detect as hard-coded strings all text enclosed within double quotes ("") which are used as string delimiters in Java … It would therefore be desirable to develop a scanning program that identifies non-externalized strings … that are not hard-coded but that are enclosed within string delimiters".[1]

[1] Excerpted from <u>Background Information</u>, pages 1-4 of the application as filed

Thus, it is submitted that one of ordinary skill would realize that the inventive process and system will help software developers to focus more quickly on actual hard-coded strings, which need to be changed, rather than on non-externalized strings that are not hard-coded and do not need changing. This rejection is now believed overcome.

II. <u>35 U.S.C. § 112, Second Paragraph</u>

Claims 1, 3, 9, 11, 17, and 19 have been rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. This rejection is respectfully traversed.

The rejection states that the structure limitation does not limit the process of steps and or actions. The rejection further notes the use of a trademarked name (e.g. Java) in the claims. It is believed that the amendments to overcome the 101 rejection and the replacement of "Java" by "platform-independent code" remove the basis of this rejection.

Therefore the rejection of claims 1, 3, 9, 11, 17, and 19 under 35 U.S.C. § 112, second paragraph has been overcome.

III. <u>35 U.S.C. § 103, Obviousness</u>

Claims 1, 2, 7-10, 15, and 16 are rejected under 35 U.S.C 103(a) as being obvious over Kushmerick et al. (US 6,304,870) in view of Bowen et al. (US 6,094,649). This rejection is respectfully traversed.

Representative claim 1 now reads,

> 1. (Amended) A method for identifying non-externalized strings that are not hard-coded comprising the computer-implemented steps of:
> scanning a code for a first pair of string delimiters;
> determining whether a string within said first pair of string delimiters is a path name to a resource file; and
> if said string is not a path name to said resource file then flagging said string as a possible hard-coded string.

It is respectfully submitted that the art relied on does not meet either the 'determining' step or the 'flagging' step above. With regard to claim 1, the office action states,

> "Kushmerick discloses:
> - scanning a code for a first pair of delimiters (col.8, line 25-28 "The wrapper searches for pairs of delimiters that indicate the beginning and end of each attribute; the page is scanned for these delimiters and the extracted text returned the end of string delimiter.)
>
> Kushmerick, does not explicitly discloses the path name is a resource file.
>
> However, Bowen discloses in an analogous computer system the path name is a resource file (col. 7, lines 45-49 "resource locators include URLs, hot links, file paths, and distinguished names, object class names, table names, and primary database key values, among others")
>
> Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of directing a path or file name to a different location as taught by Bowen in to the method of scanning code as taught by Kushmerick. The modification would be obvious because of one of ordinary skill in the art would be motivated to direct to a specific data base or directory depending on the data structure as taught by Bowen (col. 4, lines 20-28)."[2]

It is noted that <u>nowhere</u> in this rejection is there an assertion that any of the references show the step of determining whether the located string is a path name, as is claimed in the present application. Rather, the rejection cites Bowen as showing that a path name can be a resource file. It is respectfully submitted that neither Bowen nor Kushmerick disclose this step of determining, nor do they disclose the newly added step of flagging the string if it is not a path name.

The referenced portions of Kushmerick states,

> As suggested earlier, "Left-Right" (LR) wrappers operate in a very simple manner. The wrapper searches for pairs of delimiters that indicate the beginning and end of each attribute; the page is scanned for these delimiters and the extracted text returned. For instance, the ExampleWrapperLR procedure is an LR wrapper for the example clothing store site:

[2] Office action of 9/9/2004, item 11, pages 4 and 5

Bowen discloses,

> Methods and systems are provided for supporting keyword searches of data items in a structured database, such as a relational database. Selected data items are retrieved using an SQL query or other mechanism. The retrieved data values are documented using a markup language such as HTML. The documents are indexed using a web crawler or other indexing agent. Data items may be selected for indexing by identifying them in a data dictionary. The indexing agent produces an index that associates keywords with resource locators such as URLs, hot links, file paths, or distinguished names. After a user provides a keyword to a search engine interface, the index is used to obtain a resource locator that is associated with the keyword. The resource locator is used to retrieve the item's current data from the structured database. A document containing the retrieved data is then generated and provided to the user.[3]
>
> The present invention provides a method and system for supporting keyword searches of data items in a structured database, such as a relational database. One method of the invention begins with selection of at least one data item in the structured database; each selected item contains data and has a corresponding location identifier which identifies the item's location within the structured database. For instance, a relational database record may be identified by an object class name and one or more primary database key values.[4]
>
> Suitable software for implementing the invention is readily provided by those of skill in the art using the teachings presented here and programming languages and tools such as Java, Pascal, C++, C, CGI, Perl, SQL, APIs, SDKs, assembly, firmware, microcode, and/or other languages and tools.[5]

It is submitted that these references only show finding a delimited string and that a resource locator can be a URL, file path, etc.; the references do not show determining <u>if</u> the string is a path name. Likewise, the cited sections do not show the added step of flagging a string if it is not a path name. Therefore, it is believed that this rejection is overcome.

[3] Bowen, abstract
[4] Bowen, column 4, lines 20-27
[5] Bowen, column 7, lines 45-49

Claims 3-6 and 11-14 are rejected under 35 U.S.C 103(a) as being obvious over Kushmerick in view of Bowen and applicant's discussion in the background section of Java, uniform resource locators, resource bundles, and dot delimited notation. This rejection is respectfully traversed.

It is submitted that the claims in this rejection are all dependent on claims in the earlier section and that as the independent claims were shown to be patentable, so too are these dependent claims. It is also noted that applicants do not assert that they have invented Java, URLs, resource bundles, or dot delimited notation; only that the specific use of these resources in this particular instance is patentable. This rejection is overcome.

Claims 17-24 are rejected under 35 U.S.C 103(a) as being obvious over Kushmerick in view of Bowen and Bell (US 6,275,978). This rejection is respectfully traversed.

Representative claim 17 recites,

> 17. (Amended) A data processing system, comprising:
> a processor; and
> a memory unit for storing instructions of said processor;
> an input mechanism;
> an output mechanism;
> a bus system for coupling the processor to the memory unit, input mechanism, and output mechanism;
> means for scanning a code for a first pair of string delimiters;
> means for determining whether a string within said first pair of string delimiters is a path name to a resource file; and
> means for flagging said string as a possible hard-coded string if said string is not a path name to said resource file.

It is submitted that the arguments above regarding Kushmerick and Bowen are also applicable to this rejection. Neither of these references shows a means for determining whether the string is a resource file; neither do they show a means for flagging a string if it is not a path name. It is further submitted that Bell does not make up this lack. Bell is directed to providing "An apparatus and method [for] providing flexible message differentiation of localized terms utilizing a resource bundle generator" and discusses a means for clarifying an ambiguous term that can be translated in several ways. However, this patent does not appear to disclose a "means for determining whether a string … is a path name"; neither does it appear to disclose a "means for flagging said string as a possible hard-coded string". Rather, Bell is cited only for disclosing "the system with a processor, a memory, an input, an output mechanism, and a bus system coupling processor to memory unit, input mechanism, and output mechanism"[6] Thus, these patents do not show all the limitations of the instant claim.

Therefore, the rejection of claims 17-24 under 35 U.S.C. § 103 has been overcome.

IV. <u>Objection to Claims</u>

Claims 3, 11, and 19 were objected to for using a trademarked name, rather than the generic terminology. It is noted that these claims have been amended to replace "Java" with "platform-independent code". It is submitted that by this amendment, the objection to the claims have been overcome

[6] Office action of 10/09/2004, page 9, item 13

V. **Conclusion**

It is respectfully urged that the subject application is patentable over Kushmerick, Bowen, Bell, and the disclosure regarding Java and is now in condition for allowance. The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: Dec. 9, 2004

Respectfully submitted,

Betty Formby

Betty Formby
Reg. No. 36,536
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Agent for Applicants